



TC GAMING

TC-Gaming (Transfer Multi Wallet) Integration Guide

天成游戏轉換多錢包接入接口指南

TC GAMING

版本历史纪录 Version History

版本号	更新日期	撰写人员	
1.0	2015-09-29	Jack	Initial version
1.1	2016-09-16	Rex	Draft version
1.2	2016-10-05	Rex	Update on new Get Bet Detail API (AG only)
1.3	2016-10-22	Jack	Update API: Get Bet Transactions & Get Bet Transactions By Member and Get Bet Details product support information. Add a new Product Code: NT
1.4	2016-01-24	Jack	LAUNCH GAME API 5.1.6 (Support H5)
1.5	2017-02-06	Jack	Updated Lotto API
1.6	2017-02-12	Jack	Updated Realtime API and Batch API
1.7	2017-02-16	Jack	Updated Get PVP Transactions
1.7u1	2017-02-22	Phoenix	Updated API reference_no need to change to ref_no
1.7u2	2017-02-28	Phoenix	Updated API Format
1.7u3	2017-03-06	Phoenix	Add new Product Code : EA, PNG, Habanero
1.7u4	2017-03-17	Phoenix	Add LAUNCH GAME API method view
1.7u5	2017-03-30	Phoenix	Add Update Password Return Code 22
1.7u6	2017-04-07	Phoenix	Update 3.1 and 3.2 support Games
1.7u7	2017-04-10	Phoenix	Update RNG Game Desc
1.8	2017-04-17	Phoenix	Add new PRODUCT TYPE Code 14 and 15 add Game List and Game Icons add Game List API add Player Game Rank API
1.8u1	2017-05-05	Phoenix	AG support https, Update Lotty Info
1.8u2	2017-05-18	Kingsley	Update 6.3,6.4 for lotto
1.8u3	2017-05-26	Phoenix	add GAME LIST API game_type PVP
1.8u4	2017-07-18	Phoenix	Add GET RNG BET DETAILS BY MEMBER , param page Change Account Size 4 to 14
1.9	2017-08-24	Phoenix	Add New Product Ameba(AE) and CQ9 Add Transaction Type Info and add txTypeId type Add Agile Bet and Deal and Slot Add Sports API
2.0	2017-09-01	Phoenix	API Document re-layout 接口文件重新排版 Delete API Document (the API No maintained) 从文档删除接口 – 已不再提供任何维护 [PVP API] Get PVP Game Base Record 取得棋牌游戏的纪录 [PVP API] Get PVP Game Base Record By Member 取得棋牌游戏会员交易明细记录 [PVP API] Get PVP Game Detail Record 取得棋牌游戏的明细记录 [Lotto API] Get Lotto Transactions 取得彩票交易 [Lotto API] Get Lottery Betting History 取得彩票投注历史 [Lotto API] Get Lotto Transactions By Member 取得会员彩票交易明细 [Lotto API] Get Lottery Betting History By Member and Criteria 取得会员彩票投注明细 [Lotto API] Get Lottery Betting Detail 取得彩票投注明细 [Lotto API] Cancel Order 取消订单 [common API] Player Game Rank API 游戏排行榜

Add Call API the sample Code 添加每个 API 呼叫范例

Update all api response error_message to error_desc 更换所有回传值 error_message 成 error_desc

添加 API 呼叫是否为即时级批次的标示

添加 API 接口

[Lotto API]Get Lottery Order History by Member 取得会员彩票历史订单

[Lotto API]Get Lottery Settled Order History 取得彩票已完成的订单记录

[Lotto API]Get Lottery Cancelled Order Detail History 取得彩票取消订单详细历史

[Lotto API]Get Lottery Unsettle Order History 取得彩票未完成的订单历史

[Sports API]Get Sports Bet Detail By Member 取得会员体育游戏的投注明细

[PVP API]Get PVP Transactions By Member 取得会员对战游戏的交易资讯

[PVP API]Get PVP Transactions 取得对战游戏的交易资讯

2.0u1	2017-09-11	Phoenix	在游戏列表处补上游戏分类对照表
2.0u2	2017-09-26	Phoenix	添加產品六合彩代碼
2.0u3	2017-11-16	Phoenix	添加裝置分類於附件 Lottery Page Code Agile 系列不允许直接接入 · 修改产品代码
2.0u4	2017-12-07	Phoenix	添加 PVP Game List API 删除对战游戏列表
2.0u5	2017-12-18	Phoenix	增强 Overview 的说明
2.0u6	2017-12-20	Phoenix	添加 eLotto 产品资讯 添加 Launch Game eLotto 配置
2.0u7	2017-12-21	Phoenix	添加 4.1 get lotto game list 属性 lotto_type 修改附件产品代码的 EA 及 OPUS 类别属性從 RNG 换成 LIVE
2.0u8	2017-12-28	Phoenix	增加 Launch Game platform 参数 html5-desktop
2.0u9	2018-01-02	Phoenix	修改 Fund Transfer Exception 转账异常处理流程
2.0u10	1018-01-11	Phoenix	修改 lottery_bet_mode 添加 加解密說明
2.0u11	2018-01-23	Phoenix	添加 回传值有效投注 EA 修改 附件 Page Code 修改 附件彩票游戏代码 新增 天彩(视频彩)于附件彩票游戏代码
2.0u12	2018-01-31	Phoenix	改排版
2.0u13	2018-02-12	Phoenix	Add New Porduct TGP, DG, AB 添加新产品 TGP, DG, AB
3.0	2018-02-22	Alice	API document split 文档拆分
3.0u1	2018-03-05	Alice	Add New Lotto Group PCB PC 蛋蛋 新增彩种组 PCB PC 蛋蛋
3.0u2	2018-04-13	Alice	Add Language for launch game 启动游戏新增语言代码
3.0u3	2018-05-21	Alice	Add New Elotto launch game Sample, New Lotto Group K3 快3 新增 Elotto 启动游戏 sample, 彩种组 K3 快3
3.0u4	2019-01-11	Alice	Add Gamelistr API retrnd parameters 新增游戏列表接口返回

目录 Table of Contents

Proprietary Notice 专有声明	5
1. Overview 概述	6
1.1. PROTOCOL 通讯协议	6
1.2. Encryption Keys 加密协议	6
1.3. REQUEST AND RESPONSE FORMAT 请求与响应的格式	7
1.4. API METHOD TO BE CATEGOTIES 接口方法的分类	11
2. COMMON API 通用接口	12
2.1. Create / Register Player API 创建/确认玩家接口	12
2.2. Update Password API 更新密码接口	14
2.3. Get Balance API 获取余额接口	16
2.4. Fund Transfer API 资金转账接口	18
2.5. Check Transaction Status API 检查交易状态接口	20
2.6. Launch Game API 启动游戏接口	22
2.7. Game List API 游戏列表接口	38
3. Contact information 联系方式	42

Proprietary Notice 专有声明

The information contained in this document is subject to change with the Change Request procedures as specified in the Project Agreement.

TC-Gaming Company shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Copyright. Including software, file formats, and audiovisual displays; may be used pursuant to applicable software license agreement; contains confidential and proprietary information of TC Gaming and/or third parties which is protected by copyright law, trade secret law, and international treaty, and may not be provided or otherwise made available without proper authorization.

©Copyright 2017 TC-Gaming Group.

本文档中包含的信息可能会根据项目协议中规定的更改请求程序进行更改。

天成游戏不对本文包含的错误或与本材料的提供、执行或使用相关的偶然或必然损失负责。

版权.包括软件、文件格式和视听显示; 可根据适用的软件许可协议使用; 包含受版权法、商业秘密法和国际条约保护的天成游戏和/或第三方的机密和专有信息, 未经适当授权, 不得提供或以其他方式提供。

©Copyright 2017 天成游戏.

TC GAMING

1. Overview 概述

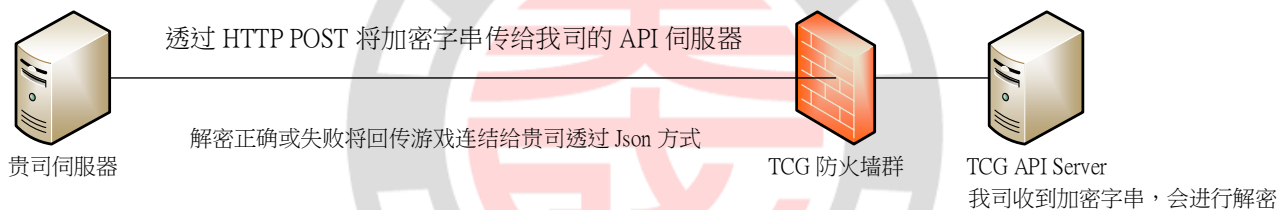
TC Gaming provides this API integration document to software, consulting services and API configurations related to TC-Gaming products.

天成游戏提供了这个接口集成文档、软件、咨询服务和与天成娱乐产品相关的接口配置。

1.1. PROTOCOL 通讯协议

The network communication will be sent as HTTP POST, with request in POST params and result in JSON format.

网络通讯使用 HTTP POST 发送，结果以 JSON 格式返回。



贴心提醒：关于我司 API 防火墙每一条线路 (品牌) 上线前有 5 个 IP 为限制，上线后有 10 个。超过 部分会产生费用，费用详情请联系我们家业务，或者帐单上会显示哟！

1.2. Encryption Keys 加密协议

Parameters are encrypted using DES encryption ECB mode with PKCS5 Padding and encoded with Base64.

SHA-256 with 64 char hexadecimal strings is used for signing the params to ensure that no tampering happens.

参数使用 DES 加密 ECB 模式加密，使用 PKCS5 填充并使用 Base64 编码。

具有 64 字符十六进制字符串的 SHA-256 用于签署参数，以确保不会发生篡改。

1.3. REQUEST AND RESPONSE FORMAT 请求与响应的格式

HttpClient timeout should be set greater than or equal to 30 seconds.

Http Client 超时时间需大于等于 30 秒。

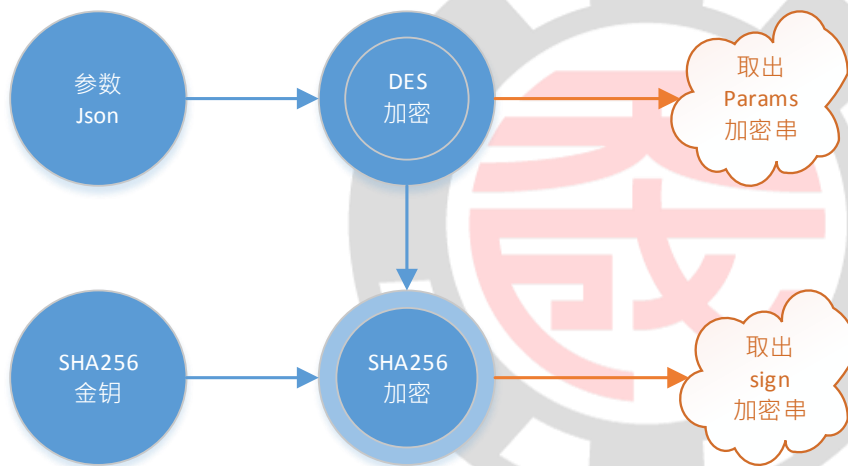
接口网址 (End Point) : `http://<server domain/ip> /doBusiness.do`

方法 (Method) : POST

类型 (Type) : REST

<server domain/ip> please check online detail file

伺服器网址/位置 请参考上线文档



说明：

首先先把所需要呼叫的方法组成 **Json**，透过开线文档中提供的 **DES** 加密金钥加密，加密后为 **params** 参数，在把此参数及 **SHA256** 金钥加密，成为 **sign** 参数，透过 **HTTP POST** 将 商户号及 **params** 参数及 **sign** 传送给我司。

`http://<server domain/ip> /doBusiness.do? merchant_code=代理商号¶ms=透过 DES 加密后的 params 字串 &sign=透过 SHA256 加密后的字串`

Parameters 参数

Parameter Name 参数名称	Parameter Value 参数值
merchant_code	assigned merchant code 代理商号
params	encrypted json string (each api will have different json parameter format) parameter using DES 使用 merchant_des_key 通过 DES 加密 JSON 串(每个接口有不同个数的参数)

	E.G 范例 <pre>DES ({ param1: value1, param2: value1, })</pre> with the merchant des key given 请参考上线文档的金钥
sign	SHA256(params + merchant_hash_key)

Response Format (响应的格式) : JSON, UTF-8

※Please transcode to UTF-8, otherwise server will receive garbled message, which can not be parse.

※烦请转码为 UTF-8，否则伺服器接收到的讯息会为乱码，无法解析。

Java 加密流程范例：

```
/**
 * 呼叫天成接口
 * @param json 参数
 * @return 结果字符串
 */
protected String doRequest(String json){

    try {
        // 参数加密
        DESEncrypt des = new DESEncrypt(config.getDesKey());
        String encryptedParams = des.encrypt(json);

        //签名档加密
        String sign = HashUtil.sha256(encryptedParams+config.getSignKey());

        //组连接字符串
        String data = "merchant_code="+URLEncoder.encode(config.getMerchantCode(),"UTF-8")
            + "&params="+URLEncoder.encode(encryptedParams,"UTF-8")
            + "&sign="+URLEncoder.encode(sign,"UTF-8");

        //传送
        return HttpUtils.newPost(config.getApiUrl(), data).execute();
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}
```


C#.NET 加密流程范例：

在范例中我添加参考：Newtonsoft.Json.dll 、System.Web

Newtonsoft.Json.dll 档案在 debug 资料夹下有，而 System.Web 可能您要修改专案目标为 .NET Framework 4 版本才有。

```
//呼叫天成接口
public String doRequest(string json)
{
    //明文
    decTextBox.Text = json;

    //参数加密
    DESCrypt des = new DESCrypt(config.desKey);
    String encryptedParams = des.encrypt(json);
    JsonTextBox.Text = encryptedParams;

    //签名档加密
    String sign = HashUtil.sha256(encryptedParams + config.signKey);
    SignTextBox.Text = sign;

    //组连接字符串
    String data = "merchant_code=" + HttpUtility.UrlEncode(config.merchantCode)
        + "&params=" + HttpUtility.UrlEncode(encryptedParams)
        + "&sign=" + HttpUtility.UrlEncode(sign);
    encTextBox.Text = data;

    return HttpUtilities.HttpPost(config.apiUrl, data);
}
```

VB.NET 加密流程范例：

在范例中我添加参考：Newtonsoft.Json.dll 、System.Web

Newtonsoft.Json.dll 档案在 debug 资料夹下有，而 System.Web 可能您要修改专案目标为 .NET Framework 4 版本才有。

```
'呼叫天成接口
Public Function doRequest(ByVal json As String) As String
    '明文
    decTextBox.Text = json

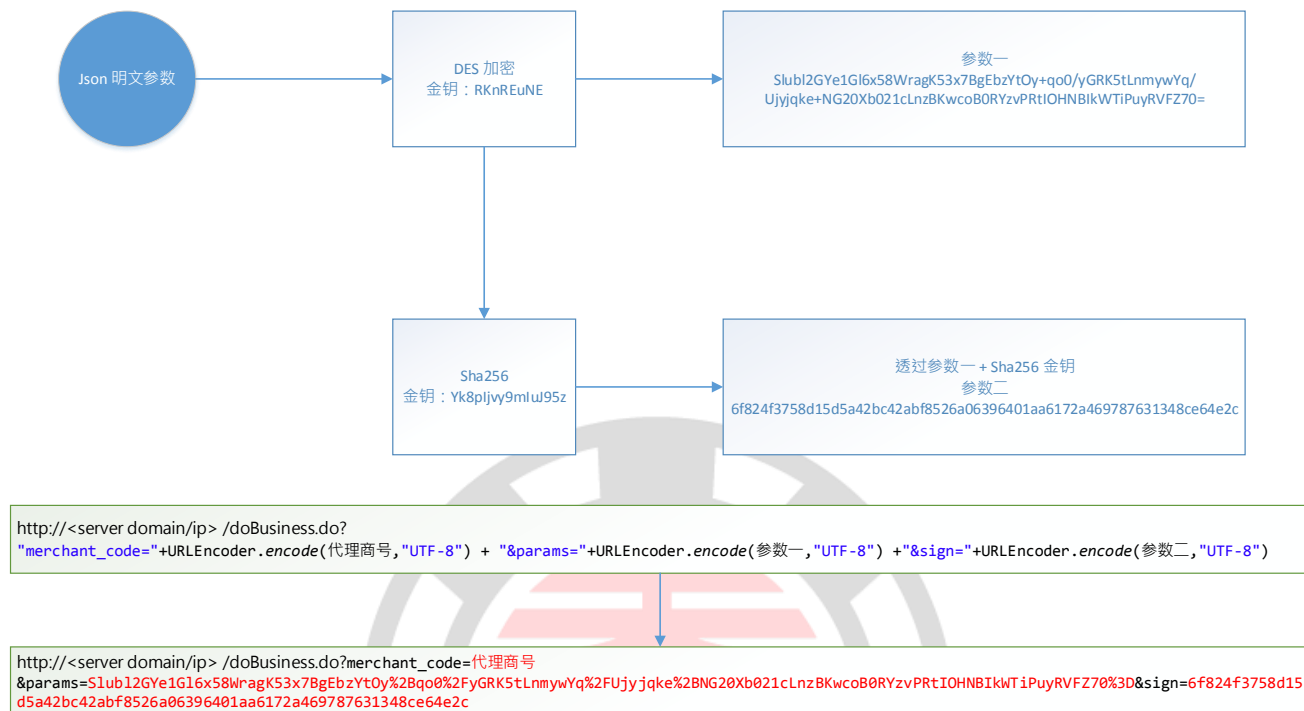
    '参数加密
    Dim des As DESCrypt = New DESCrypt(Me.config.desKeyDo)
    Dim encryptedParams As String = des.encrypt(json)
    JsonTextBox.Text = encryptedParams

    '签名档加密
    Dim sign As String
    sign = HashUtil.sha256((encryptedParams + Me.config.signKeyDo))
    SignTextBox.Text = sign

    '组传送字符串
    Dim data As String = ""
    data = "merchant_code=" + HttpUtility.UrlEncode(Me.config.merchantCodeDo) +
        "&params=" + HttpUtility.UrlEncode(encryptedParams) +
        "&sign=" + HttpUtility.UrlEncode(sign)

    encTextBox.Text = data

    Return HttpUtilities.HttpPost(Me.config.apiUrlDo, data)
End Function
```



若您要确认您的加密参数是否正确，可以透过下面资讯确认 (下面内容是以做了 java encode utf8 或 C# , VB.Net 的 Urlencode 结果)

Parameter Name 参数名称	Parameter Value 参数值
DesKey	RKnREuNE
SignKey	Yk8pljvy9mIuJ95z
Json	{"method": "cm", "username": "phoenixTest", "password": "1111qqqq", "currency": "CNY"}
Code Version	Java
params	S1ub12GYe1G16x58WragK53x7BgEbZyT0y%2Bqo0%2FyGRK5tLnmywYq%2FUjyjqke%2BNG20Xb021cLnzBKwcoB0RYzvPRtIOHNBikWTiPuyRVFZ70%3D
sign	6f824f3758d15d5a42bc42abf8526a06396401aa6172a469787631348ce64e2c
Code Version	VB .Net
params	S1ub12GYe1G16x58WragK53x7BgEbZyT0y%2bqo0%2fyGRK5tLnmywYq%2fUjyjqke%2bNG20Xb021cLnzBKwcoB0RYzvPRtIOHNBikWTiPuyRVFZ70%3d
sign	6f824f3758d15d5a42bc42abf8526a06396401aa6172a469787631348ce64e2c
Code Version	C# .Net
params	S1ub12GYe1G16x58WragK53x7BgEbZyT0y%2bqo0%2fyGRK5tLnmywYq%2fUjyjqke%2bNG20Xb021cLnzBKwcoB0RYzvPRtIOHNBikWTiPuyRVFZ70%3d
sign	6f824f3758d15d5a42bc42abf8526a06396401aa6172a469787631348ce64e2c

1.4. API METHOD TO BE CATEGOTIES 接口方法的分类

1. Common API (Core/Base APIs that are mandatory to be implemented to run TCG Products)
2. RNG API (All Slot Game/RNG Game Related API)
3. LOTTO API (All Lotto Game Related API)
4. PVP API (All PVP/P2P Game Related API)
5. SPORTS API (All SPORTS Game Related API)
6. ELOTTO API (All eLotto Game Related API)

1. 通用接口 (核心/基本的 API 来启动 TCG 的产品)
2. 电子/真人游戏接口 (所有的老虎机游戏/电子/真人游戏的接口)
3. 彩票接口 (所有彩票游戏的接口)
4. 棋牌游戏接口 (所有 PVP/P2P 游戏接口)
5. 体育接口 (所有体育游戏的接口)
6. E 彩票接口 (所有 E 彩票游戏的接口)

TCGAMING

2. COMMON API 通用接口

Common API are the Core/Base APIs that are mandatory to be implemented to run TCG Products.

通用接口是呼叫天成产品都一定会要的呼叫的接口。

2.1. Create / Register Player API 创建/确认玩家接口

This API is used to register a member to the TCG platform only. Member will be automatically registered in the product during the invocation of other API' s. If user already exists it will return success.

Otherwise other API' s cannot be used.

此接口仅用于向 TCG 平台注册玩家账号，如果该账号已存在，则返回成功，如果不存在则会注册一个新的游戏账号在 TCG 平台，当注册成功后，调用其他的 API 接口时，TCG 会自动注册账号到游戏产品平台（PT 账号，AG 账号等）。

Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字符串	parameter = 'cm' is a constant 数值 cm 是一个常数
username	Y	String 字符串	merchant player username (min 4 chars , 14 chars max) must be at least 4 up to 14 characters and 0-9, a-z 游戏账号的登录名(4 ~ 14 位) 不能使用特殊字符，由 4 到 14 位数字或小写字母组成
password	Y	String 字符串	merchant player password (min 6 chars , 12 chars max) The special characters are not allowed. Password must be at least 6 up to 12 characters and 0-9, A-z, 游戏账号的密码(6 ~ 12 位) 不能使用特殊字符，由 6 到 12 位数字或大小写字母组成

Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 = 成功，则请参考附件 接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件 接口回传结果表

Sample 范例：

params	<pre>{ "username" : "phoenix" , "password" : "1q2w3e4r" , "currency" : "CNY" , "method" : "cm" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null }</pre>

TC GAMING

2.2. Update Password API 更新密码接口

This API is use to change the password of the player. Once password is successfully updated in TCG platform, updated password will be in synchronize with the product respectively. If updating password to the product is failed, retry function in back office is provided to the operator.

此 API 用于更改玩家的密码。一旦在 TCG 平台中成功更新密码，更新密码将分别与其他游戏产品平台同步。如果产品的密码更新失败，则向操作员提供后台的重试功能。

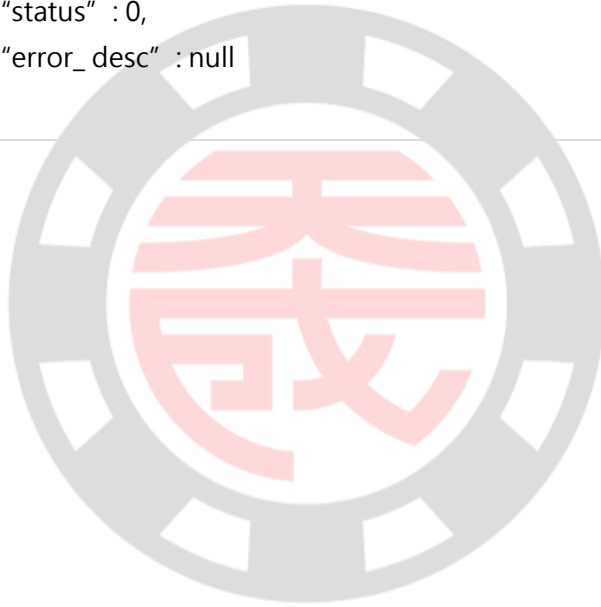
Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字符串	parameter = 'up' is a constant 数值 up 是一个常数
username	Y	String 字符串	merchant player username (min 4 chars , 14 chars max) 游戏账号的登录名(4 ~ 14 位)
password	Y	String 字符串	merchant player password (min 6 chars , 12 chars max) The special characters are not allowed. Password must be at least 6 up to 12 characters and 0-9, A-z, 游戏账号的密码(6 ~ 12 位) 不能使用特殊字符，由 6 到 12 位数字或大小写字母组成

Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 = 成功，则请参考附件 接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件 接口回传结果表

Sample 范例：

params	<pre>{ "username" : "phoenix1" , "password" : "asdf1234" , "method " : "up" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null }</pre>



TC GAMING

2.3. Get Balance API 获取余额接口

This API is use to get the member balance of each product. Operator will invoke this API for every single product subscribed.

此 API 用于获取玩家余额，只能单个产品的呼叫。

Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字符串	parameter = 'gb' is a constant 数值 gb 是一个常数
username	Y	String 字符串	merchant player username 游戏账号的登录名
product_type	Y	Int 整数	refer to the product type table 请参阅产品类型表

Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 = 成功，则请参考附件接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件接口回传结果表
balance	members balance 所查询游戏账号余额

Sample 范例：

params	<pre>{ "username" : "phoenix" , " method ": "gb" , "product_type" : 7 }</pre>
--------	---

Response

```
{  
  "status" : 0,  
  "balance" : 0,  
  "error_desc" : null  
}
```



TCGAMING

2.4. Fund Transfer API 资金转账接口

Fund transfer API cater for Fund in/Deposit and Fund out/Withdrawal in one. Unique reference number will be provided by operator. If there is any missing transaction during the transfer (E.G: connection timeout happened), operator should invoke the check transaction API to verify the status of the transaction. ([refer to check transaction status api](#))

Connection Timeout: Connection timeout should be greater or equal to 30 seconds. refer to the appendix Exception 转账异常处理

资金转账 API 用于基金/存款和资金支出/提款。唯一的参考号将由操作员提供。如果在转移期间中间有任何丢失的事务 (E.G: 发生连接超时)，运算符应调用检查事务 API 以验证事务的状态。([参考检查交易状态接口](#))

连接超时：连接超时时应大于或等于 30 秒。参阅附件转账异常处理

Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字串	parameter = 'ft' is a constant 数值 ft 是一个常数
username	Y	String 字串	merchant player username 游戏账号的登录名
product_type	Y	Int 整数	refer to Appendix 1.2 PRODUCT TYPE 请参阅附件 1.2 产品代码
fund_type	Y	String 字串	1=Fund IN/Deposit , 2 = Fund OUT / Withdraw 1 = 存款 · 2 = 提款
amount	Y	Decimal 数字	Transaction amount it should be on 2 decimal format. E.G. 12.00 交易金额 (支援到小数两位)
reference_no	Y	String 字串	transaction reference number (min 6 chars , 20 chars max) 交易参考号 (唯一值) (6 ~ 40 位)

Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 = 成功，则请参考附件 接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件 接口回传结果表
transaction_status	PENDING,SUCCESS,FAILED,UNKNOWN PENDING = 延迟，SUCCESS = 成功，FAILED = 失败，UNKNOWN = 未知

Sample 范例：

params	<pre>{ "username" : "phoenix", "amount" : 100, "method" : "ft", "product_type" : 7, "fund_type" : "1", "reference_no" : "TCGTESTDEP2017082901" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "transaction_status" : "SUCCESS" }</pre>

2.5. Check Transaction Status API 检查交易状态接口

Check transaction status is recommended when there is no response or connection timeout from the Fund Transfer API (Refer to 2.4). If the status return is failed (F), the fund transfer are required to re-invoke a new transfer with new reference number. Meanwhile if the status returned is unknown (U), this require contacting the TCG immediately for clarifications.

建议使用检查交易状态，然后若使用 [Fund Transfer API](#) 没有响应或连接超时。如果状态返回失败（FAILED），则资金转移需要重新调用具有新参考号的新转移。同时，如果状态返回未知（UNKNOWN），则需要立即联系TCG 进行问题排除。

Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字符串	parameter = 'cs' is a constant 数值 cs 是一个常数
product_type	Y	Int 整数	refer to Appendix 1.2 PRODUCT TYPE 请参阅附件 1.2 产品代码
ref_no	Y	String 字符串	transaction reference number 交易参考号 (唯一值)

Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 = 成功，则请参考附件 接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件 接口回传结果表
transaction_status	PENDING,SUCCESS,FAILED,UNKNOWN,NOT FOUND PENDING = 延迟，SUCCESS = 成功，FAILED = 失败，UNKNOWN = 未知， NOT FOUND = 未找到

Sample 范例：

params	{ "method ": "cs" ,
--------	------------------------

	<pre> "product_type" : 7, "ref_no" : "TCGTESTDEP2017082901" </pre>
Response	<pre> { "status" : 0, "error_desc" : null, "transaction_status" : "SUCCESS" } </pre>



TCGAMING

2.6. Launch Game API 启动游戏接口

This API is use to perform game login and returns the game url.

此 API 用于执行游戏登录并返回游戏网址。

Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字符串	parameter = 'lg' is a constant 数值 lg 是一个常数
username	Y	String 字符串	merchant player username 游戏账号的登录名
product_type	Y	Int 整数	refer to Appendix PRODUCT TYPE 请参阅附件产品代码
platform	N	String 字符串	Default to flash, (flash , html5, html5- desktop) *only some games supported this please refer in theAppendi tcg game list *Lotto all support html5andflash 默认为 flash · (flash · html5, html5- desktop) *注意：只有一些游戏支持这个请参考 Appendi GameList 游戏列表 *彩票都支持 html5 和 flash
game_mode	Y	String 字符串	account type of member (1 = REAL, 0 = TRIAL) 会员帐户类型 (1 =真实 · 0 =测试)
game_code	Y	String 字符串	Game code please refer to TCG Appendix : Game List. If lottery,please use get lotto game list API 游戏代码请参考 Gamelist 游戏列表 彩票:请使用获取彩票游戏列表 API

view	N	String 字符串	The page to be launch, regfer to Appendix (Lottery only) 启动的页面,请参考附件文档 (彩票使用)								
lottery_bet_mode	N	String 字符串	Only for Lottery Game 彩票游戏所必需的 <table><tr><td>Traditional</td><td>传统,</td></tr><tr><td>Traditional_Mobile_V2</td><td>传统_手机_V2</td></tr><tr><td>Elott_Traditional</td><td>e 彩票</td></tr><tr><td>Elott_Traditional_Mobil e_V2</td><td>e 彩票_手机_v2</td></tr></table>	Traditional	传统,	Traditional_Mobile_V2	传统_手机_V2	Elott_Traditional	e 彩票	Elott_Traditional_Mobil e_V2	e 彩票_手机_v2
Traditional	传统,										
Traditional_Mobile_V2	传统_手机_V2										
Elott_Traditional	e 彩票										
Elott_Traditional_Mobil e_V2	e 彩票_手机_v2										
ip_address	N	String 字符串	ip address (not used for the moment) ip 地址 (暂不使用)								
language	N	String 字符串	Default to ZH_CN 默认为 ZH_CN for other language code, please refer to Appendix1.11 Language Code 语言代码 其他语言请参阅 Appendix1.11 Language Code 语言代码								
series	N	JSON	Only for Tradition Lottery Game ,no need with Elotto , This is required for Product Type = 2 传统彩票时必填, Elotto 不需 [{ "game_group_code" : "SSC" , "prize_mode_id" : "1" , "max_series" : "1956" , "min_series" : "1700" , "max_bet_series" : "1950" , "default_series" : "1800" },] 目前提供的彩种组如下：								

			时时彩:SSC 低频彩:LF 11 选 5:11X5 六合彩:LHC PK10:PK10 PC 蛋蛋:PCB 快三:K3
--	--	--	--

参数	说明
game_group_code	Game group code (ex : SSC) 彩种组代码 (详见附件请看 LOTTERY GAME CODE 彩票游戏代码)
prize_mode_id	1 for traditional mode , 2 for ZY mode 奖金模式 · 1 為傳統
max_series	The max series of the game group. (ex : 1956) 彩种組最高奖金系列
min_series	The min series of the game group.(ex : 1700) 彩种組最低奖金系列
max_bet_series	The max series user can choice to bet.(ex : 1950) 该彩种組最大可投注系列 · 请填写 max_series 相同值
default_series	The default series to bet.(ex : 1800) 该彩种組预设投注系列 · 请填写 min_series 相同值

举例：

奖金 = 单注投注金额 x 奖金系列 / 标准奖金 x 投注模式(元角分厘) x 赔率

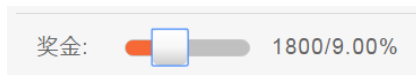
以时时彩三星直选为例(全选 2000 , 中奖概率 1/1000 赔率 1000, 所以玩家赚 1956)

min_series : 假设玩家全选花了 2000 奖金就是 1956

min_series : 如果现在系列是 1700 三星直选时 · 花 2000 就陪 1700

max_bet_series : 此参数作为配额活动 · 假设奖金系列 1956 的帐号 · 投注系列指能选择 1950 · 那们三星直选为例 · 他中奖可以中奖最高 1950 · 他可以拿 0.3 % (1956-1950)/2000 投注返点。

Default_series : 默认显示系列值拉杆。



Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 = 成功，则请参考附件 接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件 接口回传结果表
game_url	game launch url 游戏连接
pt_username	credentials for PT login (use for game client) PT 游戏客户端登入帐号
pt_password	credentials for PT login (use for game client) PT 游戏端登入密码

PVP Sample 对战游戏范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 1, "game_code" : " 070001" , (此部份 GameCode 请看 附件 PVP 游戏列表) "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://web.uxgaming.com/MainWeb.html?cust*****" }</pre>

Elotto Sample E 彩票网页范例：

param	<pre>{ "method": "lg", "username": "phoenix", "game_code": "Lobby", "game_mode": "1", "platform": "html5", "lottery_bet_mode": "Elott_Traditional", "view": "Lobby", "product_type": 2 }</pre>
Response	<pre>{ "status": 0, "error_desc": null, "error_message": null, "game_url": "lotto/t2/?token=40145c5d-e56d-4e6b-8a68-f3a9818eeaec&merchant=tcgeb2b&lang=zh_CN&css=alasddev.css&prize_mode=Elott" }</pre>

Elotto Sample E 彩票手机网页范例：

param	<pre>{ "method": "lg", "username": "phoenix", "game_code": "game_list", "game_mode": "1", "platform": "html5", "lottery_bet_mode": "Elott_Traditional_Mobile_V2", "view": "game_list", "product_type": 2 }</pre>
-------	--

Response	<pre>{ "status": 0, "error_desc": null, "error_message": null, "game_url": "m/lotto-h5/t1/game_list?token=2d536fdb-2869-4a59-b9f8- d54c5bed1d99&merchant=tcgeb2b&css=alasddev.css&prize_mode=Elott" }</pre>
----------	---

LOTTO Web Sample 彩票网页版范例：

param	<pre>{ "username" : "phoenix" , "method " : "lg" , "product_type" : 2, "game_code" : "Lobby" , "game_mode" : "1" , "platform" : "html5" , "lottery_bet_mode" : "Traditional" , "view" : "Lobby" , "series" : [{ "game_group_code" : "SSC" , "prize_mode_id" : 1, "max_series" : 1980, "min_series" : 1700, "max_bet_series" : 1980, "default_series" : 1700 },{ "game_group_code" : "11X5" , "prize_mode_id" : 1, "max_series" : 1980, "min_series" : 1700, "max_bet_series" : 1980, "default_series" : 1700 }] }</pre>
-------	---

```

    },{
      "game_group_code" : "LF" ,
      "prize_mode_id" : 1,
      "max_series" : 1980,
      "min_series" : 1700,
      "max_bet_series" : 1980,
      "default_series" : 1700
    },{
      "game_group_code" : "PK10" ,
      "prize_mode_id" : 1,
      "max_series" : 1980,
      "min_series" : 1700,
      "max_bet_series" : 1980,
      "default_series" : 1700
    },{
      "game_group_code" : "PCB" ,
      "prize_mode_id" : 1,
      "max_series" : 1980,
      "min_series" : 1700,
      "max_bet_series" : 1980,
      "default_series" : 1700
    },{
      "game_group_code" : "K3" ,
      "prize_mode_id" : 1,
      "max_series" : 1980,
      "min_series" : 1700,
      "max_bet_series" : 1980,
      "default_series" : 1700
    }
  ]
}

```

Response

```

{
  "status" : 0,
  "error_desc" : null,
  "game_url" : "lotto/t1/?token=8e***** &css=CSS"
}

```

	}		
使用说明	<p>贵司 MIF Form 指向域名的连接 + game url</p> <p>TCG Lottery 天成彩票 (范例)</p> <table border="1"> <tr> <td>Lottery 指向域名</td><td>https://lotto.tc-gaming.com</td></tr> </table> <p>范例：https://lotto.tc-gaming.com/lotto/t1?token=***** &css=CSS</p>	Lottery 指向域名	https://lotto.tc-gaming.com
Lottery 指向域名	https://lotto.tc-gaming.com		

LOTTO Mobile Sample 彩票手机版范例：

param	<pre> { "username" : "qq123123" , "method " : "lg" , "product_type" : 2, "game_code" : "Game_List" , "game_mode" : "1" , "platform" : "html5" , "lottery_bet_mode" : "Traditional_Mobile_V2" , "view" : "Game_List" , "series" : [{ "game_group_code" : "SSC" , "prize_mode_id" : 1, "max_series" : 1980, "min_series" : 1700, "max_bet_series" : 1980, "default_series" : 1700 }, { "game_group_code" : "11X5" , "prize_mode_id" : 1, "max_series" : 1980, "min_series" : 1700, "max_bet_series" : 1980, "default_series" : 1700 }] }</pre>
-------	--

```

    "game_group_code" : "LF" ,
    "prize_mode_id" : 1,
    "max_series" : 1980,
    "min_series" : 1700,
    "max_bet_series" : 1980,
    "default_series" : 1700
  },{
    "game_group_code" : "PK10" ,
    "prize_mode_id" : 1,
    "max_series" : 1980,
    "min_series" : 1700,
    "max_bet_series" : 1980,
    "default_series" : 1700
  },{
    "game_group_code" : "PCB" ,
    "prize_mode_id" : 1,
    "max_series" : 1980,
    "min_series" : 1700,
    "max_bet_series" : 1980,
    "default_series" : 1700
  },
{
  "game_group_code" : "K3" ,
  "prize_mode_id" : 1,
  "max_series" : 1980,
  "min_series" : 1700,
  "max_bet_series" : 1980,
  "default_series" : 1700
}
]
}

```

Response

```

{
  "status" : 0,
  "error_desc" : null,
  "game_url" : "game_list.html?token=*****"
}

```

	}		
使用说明	<p>贵司 MIF Form 指向域名的连接 + game url + &line=vip1</p> <p>TCG Lottery 天成彩票 (范例)</p> <table border="1"> <tr> <td>Lottery 指向域名</td><td>https://lotto.tc-gaming.com</td></tr> </table> <p>范例 : <a &line='vip1"' href="https://lotto.tc-gaming.com/game_list.html?*****">https://lotto.tc-gaming.com/game_list.html?*****"&line=vip1</p>	Lottery 指向域名	https://lotto.tc-gaming.com
Lottery 指向域名	https://lotto.tc-gaming.com		

PT Sample 范例 :

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 3, "game_code" : "P00045" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://*****" , (网页版直接调用此连接) "pt_username" : "TCGphoenix" , (登入电脑版客户端使用) "pt_password" : " ***** " (登入电脑版客户端使用) }</pre>

AG Sample 范例 :

param	<pre>{ "method " : "lg" , "username" : "qq123123" , "product_type" : 4, "game_code" : "A00001" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

MGS Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 6, "game_code" : "M00465" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

GG Sample 范例：

param	<pre>{ "method": "lg", "username": "phoenix", "product_type": 7, "game_code": "G00001", "game_mode": "1" }</pre>
Response	<pre>{ "status": 0, "error_desc": null, "game_url": "http://***** " (网页版直接调用此连接) }</pre>

BBIN Sample 范例：

param	<pre>{ "method": "lg", "username": "phoenix", "product_type": 8, "game_code": "BB0081", "game_mode": "1" }</pre>
Response	<pre>{ "status": 0, "error_desc": null, "game_url": "http://***** " (网页版直接调用此连接) }</pre>

TTG Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 9, "game_code" : "TT0001" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

EA Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 11, "game_code" : "EA0001" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

PNG Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 12, "game_code" : "PNG001" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

HB Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 13, "game_code" : "HB0023" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

OPUS Sample 范例：

param	<pre>{ "method ": "lg" , "username" : "phoenix" , "product_type" : 14, "game_code" : "OP0001" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

ISB Sample 范例：

param	<pre>{ "method ": "lg" , "username" : "phoenix" , "product_type" : 15, "game_code" : "IS0162" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

CQ9 Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 16, "game_code" : " CQ0002" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

Ameba Sample 范例：

param	<pre>{ "method " : "lg" , "username" : "phoenix" , "product_type" : 17, "game_code" : " AE0002" , "game_mode" : "1" }</pre>
Response	<pre>{ "status" : 0, "error_desc" : null, "game_url" : "http://***** " (网页版直接调用此连接) }</pre>

2.7. Game List API 游戏列表接口

This API is used to retrieve all TCG game list available and currently enable to the TCG system.

该 API 用于检索可用的所有 TCG 游戏列表，并且当前启用 TCG 系统。

Parameters 参数	Required 必要	Data Type 资料型态	Parameter Value 参数值
method	Y	String 字串	parameter = 'tgl' is a constant 数值 tgl 是一个常数
product_type	Y	Int 整数	refer to the product type table 请参阅产品类型表
platform	Y	String 字串	flash or html5 or html5-desktop flash, html5-desktop:电脑网页版 web version html5:手机网页版 mobile web version
client_type	Y	String 字串	pc = 电脑客户端, phone = 手机客户端, web = 电脑网页版, html5 = 手机网页版 all=所有
game_type	Y	String 字串	RNG, LIVE, PVP, FISH,SPORTS
page	N	Int 整数	if no value default to page = 1 如果没有值默认为 page = 1
page_size	N	Int 整数	page counts 每页显示几笔

Response 回传

Parameters 参数	Parameter Value 参数值
status	if 0 = Success, else refer to the appendix API Return The Result 如果 0 =成功，则请参考附件 接口回传结果表
error_desc	If params is Failed, please refer to appendix API Return The Result 如果失败才会出现此参数，参数请参考附件 接口回传结果表

Parameters 参数	Parameter Value 参数值																				
games	<pre>[{ "displayStatus": 0, "gameType": "XX", "gameName": "XXXXX", "tcgGameCode": "XXX", "productCode": "XX", "productType": "X", "platform": "XXX", "gameSubType": "XX", "trialSupport": false/true }]</pre>																				
<table border="1"> <thead> <tr> <th>参数</th><th>说明</th></tr> </thead> <tbody> <tr> <td>displayStatus</td><td>0:正常 normal 1:冻结 freeze</td></tr> <tr> <td>gameType</td><td>游戏类型 game type RNG, LIVE, PVP, FISH,SPORTS</td></tr> <tr> <td>gameName</td><td>游戏名称 game name</td></tr> <tr> <td>tcgGameCode</td><td>天成游戏代码 tcg game code</td></tr> <tr> <td>productCode</td><td>产品名称 product name</td></tr> <tr> <td>productType</td><td>产品代码(详见附件文档) product type (refer to appendix)</td></tr> <tr> <td>platform</td><td>支持平台 flash or html5 or html5-desktop</td></tr> <tr> <td>gameSubType</td><td>子目录(详见附件文档) Subtype(refer to appendix)</td></tr> <tr> <td>trialSupport</td><td>试玩 trail mode true:支持 support false:不支持 not support</td></tr> </tbody> </table>		参数	说明	displayStatus	0:正常 normal 1:冻结 freeze	gameType	游戏类型 game type RNG, LIVE, PVP, FISH,SPORTS	gameName	游戏名称 game name	tcgGameCode	天成游戏代码 tcg game code	productCode	产品名称 product name	productType	产品代码(详见附件文档) product type (refer to appendix)	platform	支持平台 flash or html5 or html5-desktop	gameSubType	子目录(详见附件文档) Subtype(refer to appendix)	trialSupport	试玩 trail mode true:支持 support false:不支持 not support
参数	说明																				
displayStatus	0:正常 normal 1:冻结 freeze																				
gameType	游戏类型 game type RNG, LIVE, PVP, FISH,SPORTS																				
gameName	游戏名称 game name																				
tcgGameCode	天成游戏代码 tcg game code																				
productCode	产品名称 product name																				
productType	产品代码(详见附件文档) product type (refer to appendix)																				
platform	支持平台 flash or html5 or html5-desktop																				
gameSubType	子目录(详见附件文档) Subtype(refer to appendix)																				
trialSupport	试玩 trail mode true:支持 support false:不支持 not support																				

Parameters 参数	Parameter Value 参数值								
page_info	<pre>{ totalPage: xxx, currentPage: xx, totalCount: xx }</pre>								
	<table border="1"> <thead> <tr> <th>参数</th><th>说明</th></tr> </thead> <tbody> <tr> <td>totalPage</td><td>总共几页</td></tr> <tr> <td>currentPage</td><td>现在第几页</td></tr> <tr> <td>totalCount</td><td>总笔数</td></tr> </tbody> </table>	参数	说明	totalPage	总共几页	currentPage	现在第几页	totalCount	总笔数
参数	说明								
totalPage	总共几页								
currentPage	现在第几页								
totalCount	总笔数								

Sample 范例：

params	<pre>{ "platform": "all", "page": 0, "method": "tgl", "product_type": "7", "client_type": "all", "game_type": "RNG", "page_size": 0 }</pre>
Response	<pre>{ "status": 0, "games": [{ "displayStatus": 0, "gameType": "RNG", "gameName": "Big fish eat small fish_大鱼吃小鱼", "tcgGameCode": "G00005", "productCode": "GG", "productType": "7", </pre>


```

    "platform": "flash,html5",
    "gameSubType": "SM",
    "trialSupport": true
  },
  {
    "displayStatus": 0,
    "gameType": "RNG",
    "gameName": "GameLobby_游戏大厅",
    "tcgGameCode": "G00008",
    "productCode": "GG",
    "productType": "7",
    "platform": "flash,html5",
    "gameSubType": "SM",
    "trialSupport": true
  },
  {
    "displayStatus": 0,
    "gameType": "RNG",
    "gameName": "Lucky five_幸运五张 ",
    "tcgGameCode": "G00007",
    "productCode": "GG",
    "productType": "7",
    "platform": "flash,html5",
    "gameSubType": "SM",
    "trialSupport": true
  },
  {
    "displayStatus": 0,
    "gameType": "RNG",
    "gameName": "Shooting Dragon Gate_射龙门",
    "tcgGameCode": "G00006",
    "productCode": "GG",
    "productType": "7",
    "platform": "flash,html5",
    "gameSubType": "SM",

```

```
"trialSupport": true
}
],
"error_desc": null,
"page_info": {
  "totalPage": 1,
  "currentPage": 1,
  "totalCount": null
}
}
```

3. Contact information 联系方式

业务支援

如有任何产品需求问题，请与我们的TCG官方业务服务团队联络

邮箱：sales@tc-gaming.com

Skype：live:sales_60283

技术支持

如有任何技术上的问题，请与我们的TCG官方接入技术服务团队联络

邮箱：psm-sse@tc-gaming.com

Skype：live:phoenix.w_1